Please cancel claims 1-4, 6-9, 12, 14-16, 18 and 29-39.

Please amend the claims as follows.

1-4) Cancelled

1   5)   (Currently amended) ~~The apparatus of claim 4~~ An apparatus comprising:

2   at least one processor;

3   a memory coupled to the at least one processor;

4   a database optimizer residing in the memory and executed by the at least one

5 processor, the database optimizer using statistics regarding the type of applications

6 accessing data in a database, the frequency with which the applications access the data,

7 and the location of the data being accessed by the applications to make at least one

8 change to the database schema to optimize the performance of accessing data in the

9 database;

10   wherein the change to the database schema comprises adding a new column of a

11 second data type to the database that contains the same data in an existing column of a

12 first data type in the database; and

13   wherein the database optimizer further comprises a data coherency mechanism for

14 maintaining data coherency between the existing column and the new column.

6-9)   (Cancelled)

1   10)   (Original) An apparatus comprising:

2   at least one processor;

3   a memory coupled to the at least one processor;

4   a database residing in the memory;

5   a database optimizer residing in the memory and executed by the at least one processor,

6   the database optimizer comprising:

7   a data access mechanism that uses statistics regarding the type of applications

8   accessing data in a database, the frequency with which the applications access the

9   data, and the location of the data being accessed by the applications to make at

10   least one change to the database schema to optimize the performance of accessing

11   data in the database;

12   customization settings that may be set by a human user to determine the function

13   of the database optimizer;

14   a data coherency mechanism that maintains coherency of reflective columns in the

15   database that are created by the data access mechanism and that contain the same

16   data in different data types; and

17   a data type conversion mechanism that converts data in a first data type retrieved

18   from the database to a second data type that is preferred by a requesting

19   application.

1   11)   (Original) An apparatus comprising:

2   at least one processor;

3   a memory coupled to the at least one processor;

4   a database residing in the memory;

5   a database optimizer residing in the memory and executed by the at least one

6   processor, the database optimizer comprising:

7   a mechanism that reads statistics regarding the type of applications

8   accessing data in the database, the frequency with which the applications access

9   the data, and the columns being accessed by the applications;

10   if the statistics indicate that a selected type of application has a number of

11   accesses to a selected column of a first data type in the database that exceeds a

12   first threshold level, the database optimizer determines whether the statistics

13   indicate that the selected type of application has a number of accesses to the

14   selected column that exceeds a second threshold level, and if so, the database

15   optimizer changes the data type of the selected column in the database;

16   if the statistics indicate that a selected type of application has a number of

17   accesses to a selected column of a first data type in the database that exceeds a

18   first threshold level, the data optimizer determines whether the statistics indicate

19   that the selected type of application has a number of accesses to the selected

20   column that exceeds a second threshold level, and if not, the data optimizer adds a

21   new column of a second data type to the database that contains the same data in

22   the selected column, the selected column and the new column being defined as

23   reflective columns because they contain the same data in different data types;

24   wherein the data optimizer detects when one of the plurality of

25   applications requests access to data in the selected column, determines the

26   preferred data type for the requesting application, determines if the data in the

27   selected column is of the preferred data type for the requesting application, and if

(claim 11 continued)

28   the data in the selected column is of the preferred data type for the requesting

29   application, returning the data in the selected column to the requesting

30   application;

31   if the data in any column reflective of the selected column is of the

32   preferred data type for the requesting application, the database optimizer returns

33   the data from the reflective column to the requesting application;

34   if the data in the selected column and in all reflective columns, if any, is

35   not of the preferred data type for the requesting application, the database

36   optimizer converts the data to the preferred data type for the requesting

37   application, and returns the converted data to the requesting application.

12-16) (Cancelled)

1   17)   (Currently amended) ~~The method of claim 16 further comprising the step of~~ A

2   computer-implemented method for optimizing a database comprising the steps of:

3   (1) determining a preferred data type for at least one of a plurality of applications

4   that access the database;

5   (2) dynamically changing a schema for the database to provide the preferred data

6   type when at least one of the plurality of applications requests access to data in the

7   database; and

8   wherein the step of dynamically changing the schema determines what change to

9   make according to:

10   2A) the type of the plurality of applications accessing data in the database;

11   2B) the frequency with which the plurality of applications access the data;

12   and

13   2C) the location of the data being accessed by the plurality of applications;

14   wherein the step of dynamically changing the schema for the database comprises

15   the step of adding a new column of a second data type to the database that contains the

16   same data in an existing column of a first data type in the database; and

17   (3) maintaining data coherency between the existing column and the new column.

   18)   (Cancelled)

1   19)   (Currently amended) The method of claim [[12]] 17 further comprising the step

2   of gathering the statistics.

1   20)   (Currently amended) A <u>computer-implemented</u> method for reading data from a

2   database comprising the steps of:

3   (1) specifying a preferred data type for at least one of a plurality of applications

4   that access the database;

5   (2) detecting when one of the plurality of applications requests access to data in

6   the database;

7   (3) determining the preferred data type for the requesting application;

8   (4) determining if the data is stored in the database in the preferred data type for

9   the requesting application;

10   (5) if the data is stored in the database in the preferred data type for the requesting

11   application, returning the data to the requesting application;

12   (6) if the data is not stored in the database in the preferred data type for the

13   requesting application, performing the steps of:

14   (6A) converting the data to the preferred data type for the requesting

15   application; and

16   (6B) returning the converted data to the requesting application;

17   (7) reading statistics regarding the type of applications accessing data in the

18   database, the frequency with which the applications access the data, and the location of

19   the data being accessed by the applications; and

20   (8) dynamically changing a schema for the database to provide the preferred data

21   type when at least one of the plurality of applications requests access to data in the

22   database.

1   21)   (Original) The method of claim 20 further comprising the step of gathering the

2   statistics.

1   22)   (Original) The method of claim 20 wherein the step of dynamically changing the

2   schema for the database comprises the step of changing the data type of at least one

3   column in the database.

1   23)   (Original) The method of claim 20 wherein the step of dynamically changing the

2   schema for the database comprises the step of adding a new column of a second data type

3   to the database that contains the same data in an existing column of a first data type in the

4   database.

1   24)   (Currently amended) A <u>computer-implemented</u> method for optimizing accesses to
2   a database comprising the steps of:
3   (1) reading statistics regarding the type of applications accessing data in the
4   database, the frequency with which the applications access the data, and the columns
5   being accessed by the applications;
6   (2) if the statistics indicate that a selected type of application has a number of
7   accesses to a selected column of a first data type in the database that exceeds a first
8   threshold level, performing the steps of:
9   (2A) if the statistics indicate that the selected type of application has a
10   number of accesses to the selected column that exceeds a second threshold level,
11   changing the data type of the selected column in the database;
12   (2B) if the statistics indicate that the selected type of application has a
13   number of accesses to the selected column that does not exceed a second
14   threshold level, adding a new column of a second data type to the database that
15   contains the same data in the selected column, the selected column and the new
16   column being defined as reflective columns because they contain the same data in
17   different data types.

1   25)   (Original) The method of claim 24 wherein the first and second threshold levels
2   may be set by a human user via a user interface.

1   26)   (Original) The method of claim 24 further comprising the step of maintaining
2   coherency of data in the selected column and the new column.

1   27)   (Original) The method of claim 24 further comprising the step of gathering the
2   statistics.

1   28)   (Original) The method of claim 24 further comprising the steps of:

2   (3) specifying a preferred data type for at least one of a plurality of applications

3   that access the database;

4   (4) detecting when one of the plurality of applications requests access to data in

5   the selected column;

6   (5) determining the preferred data type for the requesting application;

7   (6) determining if the data in the selected column is of the preferred data type for

8   the requesting application;

9   (7) if the data in the selected column is of the preferred data type for the

10   requesting application, returning the data in the selected column to the requesting

11   application;

12   (8) determining if the data in any column reflective of the selected column is of

13   the preferred data type for the requesting application;

14   (9) if the data in a reflective column is of the preferred data type for the requesting

15   application, returning the data from the reflective column to the requesting application;

16   (10) if the data in the selected column and in all reflective columns, if any, is not

17   of the preferred data type for the requesting application, performing the steps of:

18   (10A) converting the data to the preferred data type for the requesting

19   application; and

20   (10B) returning the converted data to the requesting application.

29-39)   (Cancelled)

1   40)   (Original) A program product comprising:

2   (A) a database optimizer comprising:

3   a data access mechanism that uses statistics regarding the type of

4   applications accessing data in a database, the frequency with which the

5   applications access the data, and the location of the data being accessed by the

6   applications to make at least one change to the database schema to optimize the

7   performance of accessing data in the database;

8   customization settings that may be set by a human user to determine the

9   function of the database optimizer;

10   a data coherency mechanism that maintains coherency of reflective

11   columns in the database that are created by the data access mechanism and that

12   contain the same data in different data types; and

13   a data type conversion mechanism that converts data in a first data type

14   retrieved from the database to a second data type that is preferred by the

15   requesting application; and

16   (B) computer-readable signal bearing media bearing the database optimizer.

1   41)   (Original) The program product of claim 40 wherein the computer-readable

2   signal bearing media comprises recordable media.

1   42)   (Original) The program product of claim 40 wherein the computer-readable

2   signal bearing media comprises transmission media.

1 43) (Original) A program product comprising:

2   (A) a database optimizer comprising:

3   a mechanism that reads statistics regarding the type of applications

4   accessing data in the database, the frequency with which the applications access

5   the data, and the columns being accessed by the applications;

6   if the statistics indicate that a selected type of application has a number of

7   accesses to a selected column of a first data type in the database that exceeds a

8   first threshold level, the database optimizer determines whether the statistics

9   indicate that the selected type of application has a number of accesses to the

10   selected column that exceeds a second threshold level, and if so, the database

11   optimizer changes the data type of the selected column in the database;

12   if the statistics indicate that a selected type of application has a number of

13   accesses to a selected column of a first data type in the database that exceeds a

14   first threshold level, the data optimizer determines whether the statistics indicate

15   that the selected type of application has a number of accesses to the selected

16   column that exceeds a second threshold level, and if not, the data optimizer adds a

17   new column of a second data type to the database that contains the same data in

18   the selected column, the selected column and the new column being defined as

19   reflective columns because they contain the same data in different data types;

20   wherein the data optimizer detects when one of the plurality of

21   applications requests access to data in the selected column, determines the

22   preferred data type for the requesting application, determines if the data in the

23   selected column is of the preferred data type for the requesting application, and if

24   the data in the selected column is of the preferred data type for the requesting

25   application, returning the data in the selected column to the requesting

26   application;

(claim 43 continued)

27   if the data in any column reflective of the selected column is of the

28   preferred data type for the requesting application, the database optimizer returns

29   the data from the reflective column to the requesting application;

30   if the data in the selected column and in all reflective columns, if any, is

31   not of the preferred data type for the requesting application, the database

32   optimizer converts the data to the preferred data type for the requesting

33   application, and returns the converted data to the requesting application; and

34   (B) computer-readable signal bearing media bearing the database optimizer.

1   44)   (Original) The program product of claim 43 wherein the computer-readable

2   signal bearing media comprises recordable media.

1   45)   (Original) The program product of claim 43 wherein the computer-readable

2   signal bearing media comprises transmission media.